Document Type	Engineering Manual
Document number	ESS-0085848
Date	February 11, 2020
Revision	1
State	Released
Confidentiality Level	Internal
Page	1 (8)

ICS Engineering Manual

for Timestamping

	Name	Role/Title
Owner	Javier Cereijo Garcia	Timing System Engineer
Author	Javier Cereijo Garcia	Timing System Engineer

Contents

1 Overview

At European SpallationSource (ESS), the Integrated Control System (ICS) uses the MicroResearch Finland (MRF) Timing System[1] as its timing system for the ESS site. It provides 3 main functionalities: event generation, data transmission and timestamping.

1.1 Scope

In this engineering manual it is explained how to configure an Input/Output Controller (IOC) to get its timestamps from the timing system. It includes the set up for the event master (EVM), event receiver (EVR) and other Experimental Physics and Industrial Control System (EPICS) records that need to get their timestamp from the timing system.

1.2 Target audience

This document is targeted to ICS engineers and technical stakeholders of the ESS timing system. It is assumed that the target audience has a technical background in the MRF Timing System, the EPICS development, and a Linux environment.

2 System description

The timing system consists of an EVM which converts timing events and signals to an optical signal distributed through fan-out units to an array of EVRs. The EVRs decode the optical signal and produce hardware and software output signals based on the timing events. The EVM also distributes timestamping events that are used to transfer the precise real time to all EVRs.

The EVM gets the real time and a 1 pps (1 pulse per second) signal from an external clock, although for testing purposes it is also possible to generate the signals internally. The later should only be used for testing, since the clock is generated by software and is of limited accuracy, but it guarantees that all the timing modules (EVM and all connected EVRs) will have the same time and it comes closer to the production setup by allowing use of functions such as timestamping from device support.

The EVRs keep track of the clock by means of dedicated timestamping events. In the `mrfioc2` module EVR EPICS databases there is a record that gets synchronized with the EVM with the frequency of 1 Hz. In between that second, an internal EVR

[1] http://www.mrf.fi/

counter tracks progress of time, so that this records always holds the current time of the IOC, and it can be used as a timestamp source.

3 System environment

Before describing the timestamping mechanism, it is mandatory to have a proper system environment that consists of specific hardware and software. Here we will show the hardware and software lists and their set up in the ICS lab at ESS. The information shown in this chapter is used in the ICS lab at ESS.

3.1 Hardware

Table 1 shows the hardware list with the hardware used in the examples in this manual. It is possible to use different form factors than what is shown in the examples; for more information check the specific engineering manual.

Hardware	Info
MRF mTCA-EVM-300	
μTCA crate	Incl. PM, MCH
Concurrent Technologies AMC CPU	
MRF mTCA-EVR-300	
Optical cables	LC, optical 850 nm

Table 1 Hardware list.

3.2 Software

Table 2 shows the software list and its environment.

4 Engineering procedure

This chapter provides the minimal information to configure an IOC to get its timestamps from the timing system. It is not intended to show a complete deployment of the timing system, but the basic parts that need to be included in the desired IOC start-up script

Item	Version Info.
CentOS Linux	`7.7.1908`
Kernel	`3.10.0-1062.9.1.el7.x86_64`
mrf kernel module	Version: `1` / srcversion `E3290AD048B5B57D2EAA55E`
EPICS base	`7.0.3.1`
e3-req	`3.1.2`
mrfioc2	E3 module ver. `2.2.0-rc7`
devLib2	E3 module ver. `2.9.0`

Table 2 Software list and its version information.

to get timestamping.

4.1 EVM and EVR configuration

Listing 4.1 shows the basic start-up script for the EVM and EVR, with short comments for the important parts:

```
require mrfioc2,2.2.0-rc7

epicsEnvSet("IOC", "TIMESTAMP")
epicsEnvSet("DEV1", "EVM")
epicsEnvSet("DEV2", "EVR1")

epicsEnvSet("MainEvtCODE" "14")
epicsEnvSet("HeartBeatEvtCODE"   "122")
epicsEnvSet("ESSEvtClockRate"  "88.0525")

mrmEvgSetupPCI($(DEV1), "0e:00.0")
dbLoadRecords("evm-mtca-300-ess.db", "SYS=$(IOC), D=$(DEV1), EVG=$(DEV1), FEVT=$(ESSEvtClockRate), FRF=$(ESSEvtClockRate), FDIV=1")

mrmEvrSetupPCI("$(DEV2)",  "0a:00.0")
dbLoadRecords("evr-mtca-300u-ess.db", "EVR=$(DEV2), SYS=$(IOC), D=$(DEV2), FEVT=$(ESSEvtClockRate)")

var evrMrmTimeNSOverflowThreshold 100000


iocInit()

dbpf $(IOC)-$(DEV1):Enable-Sel "Ena Master"

dbpf $(IOC)-$(DEV1):1ppsInp-Sel "Sys Clk"

# Heart Beat 1 Hz:
```

```
dbpf $(IOC)-$(DEV1):Mxc7-Prescaler-SP 88052500
dbpf $(IOC)-$(DEV1):TrigEvt7-EvtCode-SP $(HeartBeatEvtCODE)
dbpf $(IOC)-$(DEV1):TrigEvt7-TrigSrc-Sel "Mxc7"

# Set-up of main event:
dbpf $(IOC)-$(DEV1):Mxc0-Prescaler-SP 6289464
dbpf $(IOC)-$(DEV1):TrigEvt0-EvtCode-SP $(MainEvtCODE")
dbpf $(IOC)-$(DEV1):TrigEvt0-TrigSrc-Sel "Mxc0"

dbpf $(IOC)-$(DEV2):EvtE-SP.OUT "@OBJ=EVR1,Code=14"
dbpf $(IOC)-$(DEV2):EvtE-SP 14

epicsThreadSleep 5
dbpf $(IOC)-$(DEV1):SyncTimestamp-Cmd 1
```

Listing 4.1 Start-up script `startup.iocsh`.

After being initialized the EVR will act as a time provider, as can be seen by checking `generalTimeReport` in the IOC shell, as shown in Listing 4.2:

```
81d1214-cslab-c-30824 > generalTimeReport
Backwards time errors prevented 0 times.

Current Time Providers:
    "EVR", priority = 50
        Current Time is 2020-01-30 15:18:58.783538.
    "OS Clock", priority = 999
        Current Time is 2020-01-30 15:18:58.783663.

Event Time Providers:
    "EVR", priority = 50
```

Listing 4.2 `generalTimeReport` on the IOC shell with an EVR.

This shows the 2 time providers for this system: the EVR and the last resort provider (OS clock). There is a fallback from higher priority providers (smaller value of priority, in this case the EVR) to lower priority providers (larger value of priority, usually the OS clock) if the higher priority ones fail.

In the case that the OS clock time is ahead of the EVR time there will be some increasing `Backwards time errors prevention` notifications in `generalTimeReport`, and the timestamps might show a strange behaviour at the startup of the IOC, while the EVR time catches up with the OS clock time. For this reason it is recommended to have all systems set with OS clock times close to each other.

4.1.1 Timestamping of records simultaneous with the timing system events

Some records should have the same timestamp as the event that trigger them, for example to correlate their value to a specific beam pulse. In this case the `$(IOC)-$(DEV):Time-I` record is set to process on arrival of the desired event and to have exactly the same timestamp as that event. The value of the `$(IOC)-$(DEV):Time-I` record is the current time expressed as a human-readable string `DATE HH:MM:SS`.

The easiest way of doing this is to modify line 15 of Listing 4.1 to add the macro `EVNT1HZ` equal to the event number that the `$(IOC)-$(DEV):Time-I` record should take its timestamp from, so that the line looks like:

```
dbLoadRecords("evr-mtca-300u-ess.db","EVR=$(DEV2), SYS=$(IOC), D=$(DEV2), FEVT=$(ESSEvtClockRate),
     EVNT1HZ=$(MainEvtCODE)")
```

where the `EVNT1HZ` macro has the value of the `MainEvtCODE` variable, 14.

Alternatively this configuration can be made after the startup of the IOC with:

```
iocuser@cslab-ccpu-crate07: ~$ caput TIMESTAMP-EVR1:Time-I.EVNT 14
Old : TIMESTAMP-EVR1:Time-I.EVNT        125
New : TIMESTAMP-EVR1:Time-I.EVNT        14
iocuser@cslab-ccpu-crate07: ~$ caput TIMESTAMP-EVR1:Time-I.INP "@OBJ=EVR1, Code=14"
Old : TIMESTAMP-EVR1:Time-I.INP         @OBJ=EVR1, Code=125
New : TIMESTAMP-EVR1:Time-I.INP         @OBJ=EVR1, Code=14
```

After this set the `.TSEL` field of the record that should have the desired timestamp to point to the .TIME field of the `$(IOC)-$(DEV):Time-I` record of the EVR by doing:

```
caput examplerecord.TSEL TIMESTAMP-EVR1:Time-I.TIME
```

Alternatively it can be added in the startup script after `iocInit()` using `dbpf` instead of `caput`.

Glossary

Term	Definition
EPICS	Experimental Physics and Industrial Control System
ESS	European Spallation Source
EVM	Event Master
EVR	Event Receiver
ICS	Integrated Control System
IOC	Input/Output Controller
MRF	MicroResearch Finland
pps	pulse per second

Document revision history

Revision	**Reason for and description of change**	**Author**	**Date**
1	First release	Javier Cereijo Garcia	February 11, 2020